ZoneMapAlt: An alternative to the ZoneMap metric for zone segmentation and classification

Romain Karpinski
Université de Lorraine - LORIA
Vandoeuvre-Lès-Nancy, France
romain.karpinski@loria.fr

Abdel Belaïd
Université de Lorraine - LORIA
Vandoeuvre-Lès-Nancy, France
abdel.belaid@loria.fr

***Abstract*—This paper proposes a new evaluation metric based on the existing ZoneMap metric. The ZoneMap method, designed to perform a zone segmentation evaluation and classification, is considered in the context of OCR evaluation. Its limits are spotted, described and a new algorithm, ZoneMapAlt (ZoneMap Alternative) is proposed to solve the identified limits while keeping the properties of the original one. To validate the new metric, experiments have been made on a dataset of scientific articles. Results demonstrate that the ZoneMapAlt algorithm provides greater details on segmentation errors and is able to detect critical segmentation errors.**

***Keywords*-Performance evaluation; OCR; ZoneMap metric; String matching**

I. Introduction

The aim of this paper is to propose a new metric to evaluate the performances of a layout extraction (or segmentation) method. Layout extraction methods are often part of a recognition system and represent a critical step for the recognition performances. Indeed, poor layout extraction performances can lead to severe recognition errors because the information is partial or truncated. Therefore, performance evaluation is an important task that allows one to be aware of a method quality. It also can be useful when creating a new method to understand the behavior and spot errors. In the context of OCR evaluations, the evaluation of the extracted layout can be used to compare different approaches.

A segmented document is composed of objects represented by geometric shapes such as polygons or rectangles or by their set of pixels. In this paper, we consider that zones and objects are the same.

Usually, comparing two different segmentations consist of matching reference objects (the ground truth) with hypothesis objects (output of the system). This matching can be hard to perform when many errors occur and especially when these errors accumulate. Figure 1 shows all basic segmentation errors that can be spotted in documents at line level. Let h be a hypothesis zone (HZ) and r be a reference zone (RZ). Let H be the set of hypothesis zones (HZs) such as $H = \{h_0, h_1, ..., h_i\}$, $i \in \mathbb{R}$ and R be the set of reference zones (RZs) such as $R = \{r_0, r_1, ..., r_j\}$, $j \in \mathbb{R}$. Regarding segmentation evaluation, a HZ h that includes several RZs is called a merge. Conversely, a RZ r that includes several HZs from H is called a split. These two types of errors can happen vertically, horizontally or both.

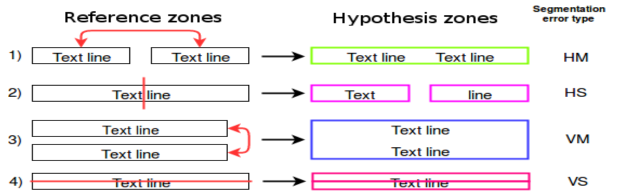
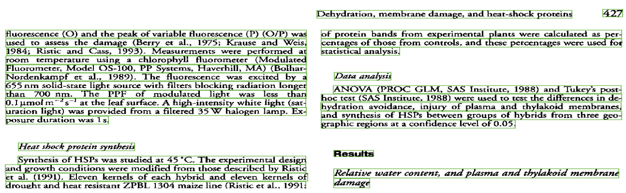
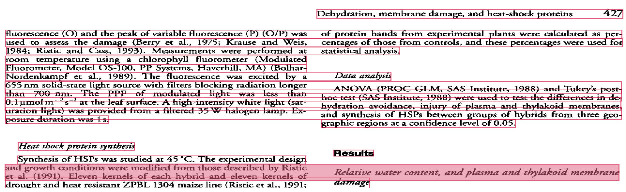


Figure 1: Segmentation errors at line level. HM is horizontal merge. HS is horizontal split. VM is vertical merge. VS is vertical split.

When comparing segmentations, several class of objects may be present such as TextLine or Graphic. Therefore, their comparison with RZs can be penalized when the class expected is not equal to the class of the match. In this case, the evaluation method assess jointly the classification and the segmentation of zones. Figure 2 illustrates two examples of segmentation, Figure 2a shows RZs and Figure 2b HZs. In this output, all zones are text lines represented by their minimum bounding box (rectangles). Since all the objects belong to the same class, the classification part of the evaluation can be ignored. In Figure 2b, as illustrated by the dark rectangle (red), merging errors occurred both vertically (joining two rows of the same column) and horizontally (joining two columns of the two rows). The evaluation of this type of segmentation is difficult because several errors occur in the same area and makes it hard to match zones in this area.



(a) Ground truth segmentation at line level



(b) System segmentation at line level

Figure 2: Example of (a) ground truth segmentation and (b) system segmentation.

Recently, the ZoneMap metric [1] has been proposed to evaluate jointly the segmentation and the classification of zones. It was developed and used for the evaluation campaign MAURDOR [2] that aimed to evaluate capabilities of methods on scanned document images. The ZoneMap algorithm has the following weaknesses:

- It does not take into account the intersection between reference zones.
- It does not permit many-to-many matching allowing to handle multiple segmentation errors.
- It groups areas that have an intersection even when their intersection area is negligible.

We propose to correct these limits in a new version of ZoneMap dubbed ZoneMapALt.

The paper is organized as follows: Section II presents methods from the state of the art on performance evaluation of zone segmentation and classification. Section III describes the ZoneMap algorithm and its limits. Section IV depicts the newly created algorithm ZoneMapAlt and shows how the ZoneMap limits are considered. In section V, experiments are performed to validate the chosen metric. Finally, section VI concludes the paper and gives perspectives.

II. Related works

Performance evaluation methods can be summarized in two categories depending on the nature of the document and ground truth availability: pixel-based or object-based. Pixel-based methods assess the class assigned to each pixel and evaluate the document segmentation either globally or sparately by object type [3]. Object-based methods seek to assess the quality of a document segmentation at a higher level (only the object and its geometric representation are considered and not its pixels). In both cases, these methods must take into account the location of pixels or objects along with their type (TextLine, Graphic, ...) and shape.

Object-based methods are often faster to compute than pixel-based methods, especially when working with high resolution images[4]. In the concern of gaining efficiency, pixel-based methods often represent pixels as polygons transformed into a set of intervals [5] [6] [7] (or uses the run-length encoding [8]). As for object-based methods, they do not consider the content of the objects (pixels) and encode only their shape represented by either polygons [9] [10] or, in more precise contexts, by rectangles [1] [4] [10]. Often, the bounding box (rectangle) is used because it is simpler to process than polygon and also represents the minimum rectangle completely including the object. Other data representation exists, such as graphs. The graphs are used to match higher level representation of pixels where nodes are segments (association of several consecutive pixels having the same class), as in [11] or an object, as in [12] where the authors use the graph probing technique. This latter technique performs the comparison by counting similar nodes or as in [13], a document model composed of a hierarchy of bounding box representations is used to perform an empirical measure of matching based on the graph-like model.

Object-based methods employs object overlap to match objects [1] [4] [9] [10] [14] [15] by utilizing the intersection area as a matching criteria. They are, therefore tributary of intersections between objects which, when they are multiple and complex, can be hard to match. In the case of an object being represented by pixel intervals, the overlap of interval pairs is used [5] [6] [7] [8] to generate the overlap matrix of intervals (matching or not) used to deduce segmentation errors.

To qualify segmentation errors, 5 types of segmentation errors which are the most frequent, as described in [9], are: noise assimilated to an object (False alarm), object from the ground truth missed (Miss), object over-segmentation (Split), object under-segmentation (Merge), simultaneous object over- and under-segmentation (Split & Merge). Liang *et al.* [10] use different terms to classify segmentation results based on relations cardinality which convey the same effects: one-to-one (Match), one-to-zero (Miss), zero-to-one (False Alarm), one-to-many (Split), many-to-one (Merge), many-to-many (Multiple errors).

The ZoneMap algorithm [1] and the PETS tool [8] consider objects classification errors in their metrics in addition to the layout assessment. They allow us both to evaluate jointly the segmentation and the object classification. Penalties are assigned to compared objects that do not have the same class. A more recent method [16] proposes to combine the object-based approach with the pixel-based approach by using a hierarchical structure of the ground truth. Objects are divided into sub-objects called elemental patterns (parts of objects) which allows one to assess them at pixel level and also to evaluate objects from the results of each of their elemental patterns.

The method we propose in this paper is object-based. It faces the same segmentation error types cited in [10]. It corrects the limits of the ZoneMap algorithm using a new reliable matching method which considers previously matched objects to avoid counting several time the same intersection area. It also allows us to assess more complex segmentation errors by enabling many-to-many associations. The ZoneMap metric and its limits are further described in Section III.

III. ZoneMap Algorithm and Limits

A. ZoneMap Algorithm

Let $L = \{h_i, r_j\}, i, j \in \mathbb{R}$ be the set of links between pairs of RZs and HZs.

The ZoneMap algorithm acts in three steps: first, force links are computed between intersecting zones, then, zones are grouped based on these links. Finally, zone groups are classified by their configuration from which the error score $E_{ZoneMap}$ is computed.

1) Computing Link Force: Links are computed for each intersecting zone between RZs and HZs. The link force between r_i and h_j is given by the equation 1:

$$Link(r_i, h_j) = (\frac{r_i \cap h_j}{r_i})^2 + (\frac{r_i \cap h_j}{h_j})^2 \quad (1)$$

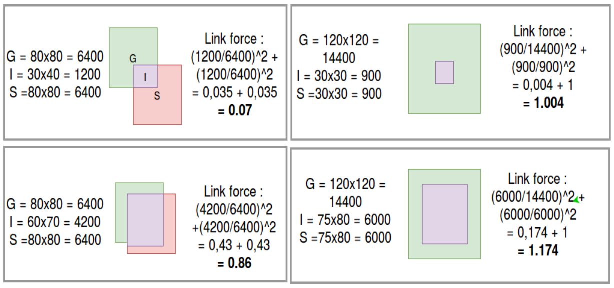


Figure 3: Example of link force values depending on the configuration of the zones.

Figure 3 illustrates the link force variation for several zones configurations.

2) Zone Grouping: The goal of the second step is to associate RZs with HZs to form groups. Only non-zero link forces are retained. Links are sorted in decreasing order of force. For each link l which associates the zones r_i and h_j: If r_i or h_j is in a group (not both at the same time) then we insert the zone that does not belong to a group into the group of the other one. If r_i and h_j do not belong to a group, we create a new group that contains both.

Inserting a zone into a group can only succeed if the final group does not contains more than one RZ and more than one HZ at the same time (i.e. many-to-many matching not allowed). If a zone could not be inserted, nothing will be done. At the end of this step, zones that do not belong to any group are considered missing if it is a RZ, or falsely detected if it is a HZ.

3) Computing the $E_{ZoneMap}$: The next step is to compute the error E_k of each group k in leading to the global error $E_{ZoneMap}$ given by the equation 2.

$$E_{ZoneMap} = \frac{100 \times \sum_{k=1}^{N} E_k}{area(R)} \tag{2}$$
$$\text{with } E_k = (1 - \alpha_c)E_s + \alpha_c E_c$$

E_s is the surface error while E_c is the classification error of the zone. $\alpha_c \in [0,1]$ is the weight given to the classification error to adjust the influence of segmentation and classification during the evaluation. The calculation of E_s and E_c depends on the group configuration.

Let H_k be a set of HZs and R_k be a set of RZs that belong to the k^{th} group. A set of groups is obtained where each group is in one of the following configurations:

1) False Alarm: The group contains only one HZ. It means this zone has no match with RZs, therefore the system has over-detected this zone.

$$E_s = area(H_k) \tag{3a}$$
$$E_c = E_s \tag{3b}$$

2) Miss: The group contains only one RZ. It means that this zone has no match with HZs, therefore the system has under-detected this zone.

$$E_s = area(R_k) \tag{4a}$$
$$E_c = E_s \tag{4b}$$

3) Match: The group contains only one HZ and one RZ.

$$E_s = area(H_k \cup R_k - H_k \cap R_k) \tag{5a}$$
$$E_c = d(t_H, t_R) \times area(H_k \cap R_k) + E_s \tag{5b}$$

$d(t_H, t_R) \in [0, 1]$ is the distance between the two zone classes.

4) Split: The group contains only one RZ and more than one HZ. It means that the RZ has been segmented into several parts. $\alpha_{MS} \in [0, 1]$ is the split/merge coefficient and $|H_k|$ is the cardinality of H_k.

$$E_s = area(H_k \cap R_k) * \alpha_{MS} * |H_k| \tag{6a}$$
$$E_c = (|H_k| - 1 + min_{h_j \in H_k} d(t_H, t_R)) \times area(H_k \cap R_k) \tag{6b}$$

5) Merge: the group contains more than one RZ and one HZ. It means that RZs have been merged.

The calculation of the fusion error is the same as for the split, but with HZ and RZ inverted. When the score $E_{ZoneMap}$ is equal to zero its means that there is no error.

A $E_{ZoneMap}$ score can be above 100 when there are a lot of *False Alarm* errors. In this case, the total erroneous surface can be superior to the sum of RZs area.

B. Limits of the ZoneMap Metric

As said in the introduction, the ZoneMap algorithm is not suited to handle complex segmentation errors (multiple split and/or merge errors). Moreover, it does not always deduce the correct type of segmentation error when there are intersections between RZs. Two limits arise. The first is about not taking into account existing reference intersections (RI_L) and assigning zones without a minimum intersection. Figure 4 illustrates the segmentation of RI_L. Zones represented by letters are RZs and zones represented by numbers are HZs. The results of the ZoneMap algorithm in Table I show that h_1 merges r_A and r_B. This error relates an association of the zones without taking into account the associations previously carried out. Regarding h_1 and r_A, their intersection is assimilated to their matching. Therefore, this intersection area cannot be used again to match with another zone for both participating zones. When trying to match h_1 with r_B, one should remove from h_1 the previously used intersection area before trying to match it with r_B. The dark area in Figure 4 corresponds to the intersection area of r_A and r_B and is responsible for the match of h_1 with r_A and r_B. The algorithm should match r_A with h_1 and inform us that r_B has been missed.

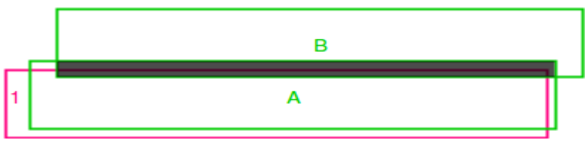


Figure 4: ZoneMap RI_L. In green, RZs. In red, HZs

Table I: Results of ZoneMap on RI_L.

Group #	Reference zone(s)	Hypothesis zone(s)	Segmentation type
1	A, B	1	Merge

The second limit describes the inability of the algorithm to perform many-to-many (MTM_L) associations. Figure 5 represents the segmentation of MTM_L. Table II contains the results for this limit and shows that h_1 matches with r_A and h_2 matches with r_B. This result is not acceptable since all zones are interconnected. The result should translate the fact that h_1 merges r_A and r_B but also segments r_A and r_B. It should be the same for h_2. However, the results produced by ZoneMap are far from the ideal error classification. The impossibility of making many-to-many associations limits the possible outcomes and therefore the accuracy of the evaluation.

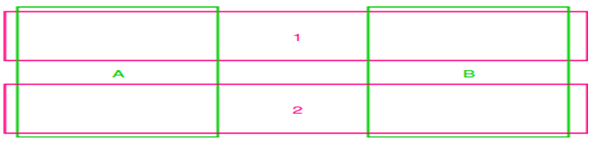


Figure 5: ZoneMap MTM_L. In green, the RZs. In red, the HZs

Table II: Results of ZoneMap on MTM_L.

Group #	Reference zone(s)	Hypothesis zone(s)	Segmentation type
1	A	1	Match
2	B	2	Match

We propose to improve the ZoneMap metric by proposing a new algorithm that handles the limits previously described. The proposed approach is reported in Section IV.

IV. ZoneMapAlt

Let $L' = \{h_i, r_j\}, i, j \in \mathbb{R}$ be the set of processed links (also called validated). The Algorithm 1 illustrates the proposed matching technique. The aim of this algorithm is to take care of intersection between RZs. This is done by considering compared zones only once. Considering RI_L of ZoneMap, the intersection between r_A and h_1 can only be counted once. Counting the intersection area only once disallows other zones to be matched with a part of a zone that has already been used in a match. Therefore, intersections between r_A and h_1 should be removed from both zones for the next matches. One important characteristic of ZoneMapAlt is its ability to perform many-to-many zone alignments. Indeed, it allows the identification and analysis of complex segmentation errors that are critical for the application.

The principle of the ZoneMapAlt algorithm is as follows ((line x) is a line reference in the algorithm):

- For each link, there are 3 main conditions:
  1) (line 7-12) If h_j is associated with at least one RZ. In this case, the union of all r_k (U_r) associated with h_j is computed. Then, the intersection of h_j with U_r is subtracted from h_j. Also, the intersection of r_i with U_r is subtracted from r_i.
  2) (line 13-17) If r_i is associated with at least one HZ. In this case, the union of all h_m (U_h) associated with the r_i is computed. Then, the intersection of r_i with U_h is subtracted from r_i.
  3) (line 18-26) If the intersection of r_i and h_j relatively to r_i is above a threshold β, the association is accepted. The link between the two polygons is marked with the cardinality of the relation (that is, the number of r_i and h_j for this relation).

For each matched zones, sub-zones have been extracted to use only intersecting areas. This can lead to remaining sub-zones that have not been used at all. These sub-zones are a *Miss* if it is a RZ and a *False Alarm* if it is a HZ. To determine if a zone (target zone) has been missed or falsely detected, one can remove successively from the target zone each zone, that been matched with it. If the resulting area is above zero, then it is a remaining zone that must be added in a single group.

A new configuration called *Multiple* that characterizes an area that has suffered several segmentation errors has been introduced. This case might happen depending on the segmentation algorithm performances and the structure of the surface analyzed. In order to heavily penalize this configuration, the erroneous surface is multiplied by a coefficient $\gamma_M \in [0; 1]$ as well as by the number of zones concerned. The surface error of a *Multiple* segmentation error zone is given by the equation 7. The classification error in this configuration is done by the equation 8

$$E_s = area(R_k \cap H_k) * \gamma_M * |R_k + H_k| \tag{7}$$

$$E_c = (|H_k + R_k| - 2 + min_{h \in H_k, r \in R_k} d(t_H, t_R)) \tag{8}$$

A user threshold $\beta \in [0, 1]$ has been introduced to determine whether the intersection is important enough relatively to the RZ. It is set empirically to 0.2 and works well on text lines. When this threshold is set to 0, the algorithm maximizes the amount of associated zones. It means that an intersection of one pixel is enough to associate two zones together. The main consideration is that a zone can be responsible for several errors and the zones with the strongest links are considered to have priority over the association.

V. Metrics Validation

A. Solving the Limiting Cases

To evaluate the ZoneMapAlt algorithm, we compared its result with ZoneMap on the two identified limits. The results obtained by ZoneMapAlt on Figure 4 and Figure 5 are shown in Table III and Table IV. All experiments were performed with the user threshold β equal to 0.2.

Regarding the results on both limits, the ZoneMapAlt provides more details on segmentation errors. In fact,

begin
Compute links L
Sort links by descending order
for $L_k \in L$ **do**
$h_j \longleftarrow L_k.h$
$r_i \longleftarrow L_k.r$
if $\exists\, r \mid \{L_k.h, r\} \in L'$ **then**
`// Remove reference association`
$U_r \longleftarrow \bigcup_{\forall r_k \mid \{L_k.h, r_k\} \in L'} r_k$
$h_j \longleftarrow h_j - (h_j \cap U_r)$
$r_i \longleftarrow r_i - (r_i \cap U_r)$
$Conf.Merge \leftarrow card(\forall r_k \mid \{L_k.h, r_k\} \in L')$
end
if $\exists\, h \mid \{h, L_k.r\} \in L'$ **then**
`// Remove hypothesis association`
$U_h \longleftarrow \bigcup_{\forall h_m \mid \{h_m, L_k.r\} \in L'} h_m$
$r_i \longleftarrow r_i - (r_i \cap U_h)$
$Conf.Split \leftarrow card(\forall h_m \mid \{h_m, L_k.r\} \in L')$
end
if $(area(h_j \cap r_i)/area(r_i)) > \beta$ **then**
if **not** *Conf.Split* & **not** *Conf.Merge* **then**
$Conf.Match \longleftarrow 1$
end
ComputeError($conf$,r_i,h_j)
$L_k \in L'$
end
end
end

Algorithme 1 : ZoneMapAlt Algorithm

Table III: ZoneMapAlt results on RI_L.

Group #	Reference zone(s)	Hypothesis zone(s)	Segmentation type
1	A	1	Match
2	B	-	Miss

for RI_L, the r_B is correctly classified as a missed zone instead of getting assigned to the group containing r_A and h_1. MTM_L is more complex, more details can be observed by ZoneMapAlt algorithm. The best match (r_A and h_1) is used as a *match* classification type then the next ones are classified regarding what classification was done previously. This allows penalizing heavily one zone that is at the end of the list so that it conveys the complexity of the segmentation error.

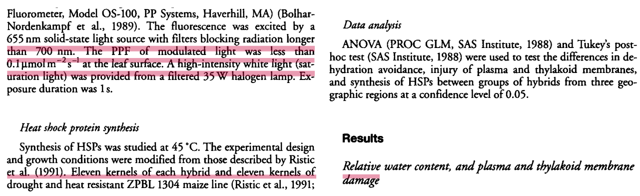
B. Real Case Example

To validate ZoneMapAlt, a real case example is discussed (see Figure 2a for its RZs and in Figure 2b for its HZs). Figure 6 shows the visual detection of the *Multiple* class. When many-to-many associations are possible, it

Table IV: ZoneMapAlt results on MTM_L.

Group #	Reference zone(s)	Hypothesis zone(s)	Segmentation type
1	A	1	Match
2	B	1,2	Split
3	A,B	2	Merge
4	A,B	1,2	Multiple

means that at least one area connects the other ones (ambiguous case) otherwise it would be a split or merge. Since configuration *Multiple* is based on many-to-many associations, it identifies areas that are responsible for the severe errors. These errors are severe because areas involved contain not only the information concerning their line but also the adjacent lines which will make it difficult to recognize.


Fluorometer, Model OS-100, PP Systems, Haverhill, MA) (Bolhar-Nordenkampf et al., 1989). The fluorescence was excited by a 655 nm solid-state light source with filters blocking radiation longer than 700 nm. The PPF of modulated light was less than 0.1 µmol m^{-2} s^{-1} at the leaf surface. A high-intensity white light (saturation light) was provided from a filtered 35 W halogen lamp. Exposure duration was 1 s.

Heat shock protein synthesis

Synthesis of HSPs was studied at 45 °C. The experimental design and growth conditions were modified from those described by Ristic et al. (1991). Eleven kernels of each hybrid and eleven kernels of drought and heat resistant ZPBL 1304 maize line (Ristic et al., 1991;

Data analysis

ANOVA (PROC GLM, SAS Institute, 1988) and Tukey's post-hoc test (SAS Institute, 1988) were used to test the differences in dehydration avoidance, injury of plasma and thylakoid membranes, and synthesis of HSPs between groups of hybrids from three geographic regions at a confidence level of 0.05.

Results

Relative water content, and plasma and thylakoid membrane damage

Figure 6: ZoneMapAlt result on a real example showing *Multiple* segmentation error class.

C. ZoneMap and ZoneMapAlt Comparison

In order to compare ZoneMap and ZoneMapAlt a common dataset is used. The goal is not to evaluate one segmentation method on a specific dataset but to compare the $E_{ZoneMap}$ differences between the two methods. This is why the dataset was created locally from INIST [1] collections. It is composed of 97 single-page and multi-column articles with headers, footnotes and references. These are documents with a very simple physical structure with few output variations. They are issued by several publishers and from several sources. They represent scientific articles in French, English, German and Spanish, from several different times (from 1942 until 2000), with different fonts. To perform the experiments, the OCRopus OCR [17] segmentation algorithm was selected.

The comparison of $E_{ZoneMap}$ scores between ZoneMap and ZoneMapAlt algorithms are shown in Table V. The main difference between the two scores is that ZoneMap classifies zones as *Missed* and *False alarmed* when ZoneMapAlt classify them as *Split*, *Merge* or *Multiple*. This can be observed with the increase of $E_{ZoneMap}$ score in the second limiting case. Indeed, for the ZoneMap algorithm on MTM_L there are two *Match* therefore remaining areas are *Missed* and *False alarm* while ZoneMapAlt produces *Match*, *Split*, *Merge* and *Multiple* with also some *Miss* and *False alarm*. The introduction of the type *Multiple* can only increase the $E_{ZoneMap}$ as in this case the surface error is greater than any other configuration types. On the other hand, as shown by the first limiting case, missed RZs that are touching other RZs will no longer be assigned to their adjacent neighbors as long as the overlapping ratio between zones (Algorithm 1 line 18) is above β. In this case, the $E_{ZoneMap}$ will decrease because a *Missed* is less severe than a *Split*.

D. Influence of β Parameter

Figure 7 shows the number of classification errors depending on the β parameter value. When $\beta = 0$, the

[1] Institute for scientific and technical information, Nancy, France

Table V: Average comparison of ZoneMap and ZoneMapAlt algorithms on 97 documents.

Method	$E_{ZoneMap}$	Match	Miss	False Alarm	Split	Merge	Multiple	Total Error
ZoneMapAlt	15,42	3 120 647	99 738	186 214	1 543	215 783	24 640	527 918
ZoneMap	14,57	3 120 647	111 926	216 526	4 484	176 033	0	508 969
ZoneMapAlt - ZoneMap	0,85	0	-12 188	-30 312	-2 941	39 750	24 640	18 949

number of matched zones is maximized. When $\beta = 1$, the number of matched zones is minimized. One can observe that when β increases the number of *Multiple* errors decreases, which means that the remaining intersection between two zones is not sufficient to associate them together. Moreover, when the number of matches decreases, the number of false alarms and misses increase. It does not increase for the same amount of error because when a zone has a missed and a match part, when the match is removed the number of miss stay the same.

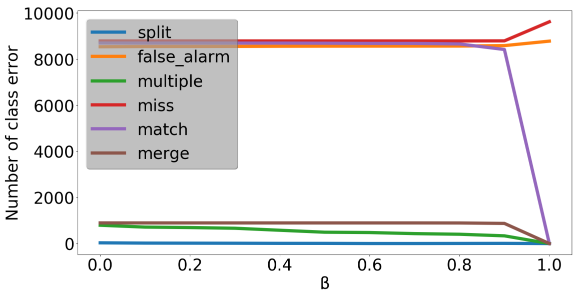


Figure 7: Number of match, miss, false alarm, split, merge and multiple errors for the whole range of value of β.

VI. Conclusion

In this paper, we proposed a new metric ZoneMapAlt based on the ZoneMap metric. This metric allows one to perform the evaluation of a segmentation algorithm. It takes into account the segmentation and classification of zones jointly. A user threshold β is introduced to control the matching sensitivity of the algorithm which makes the algorithm flexible. A new segmentation error type called *Multiple* is used to qualify the zones responsible for several errors and allows many-to-many matches. The experiments show that ZoneMapAlt provides more information on segmentation errors than ZoneMap, classifies them more accurately and is able to find zones that produce critical errors.

In the future, we plan to combine the ZoneMapAlt matching process with hierarchical pixel-based methods to allow even more precise evaluations.

References

[1] O. Galibert, J. Kahn, and I. Oparin, "The zonemap metric for page segmentation and area classification in scanned documents," in *ICIP*, 2014, pp. 2594–2598.

[2] I. Oparin, J. Kahn, and O. Galibert, "First maurdor 2013 evaluation campaign in scanned document image processing," in *ICASSP*, 2014, pp. 5090–5094.

[3] A. Antonacopoulos, B. Gatos, and D. Karatzas, "Icdar2003 page segmentation competition," *ICDAR*, pp. 688–92, 2003.

[4] M. D. Garris, "Evaluating spatial correspondence of zones in document recognition systems," in *ICIP*, vol. 3, 1995, pp. 304–307.

[5] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Scenario driven in-depth performance evaluation of document layout analysis methods," in *ICDAR*, 2011, pp. 1404–1408.

[6] F. Shafait, D. Keysers, and T. M. Breuel, "Performance evaluation and benchmarking of six-page segmentation algorithms," *PAMI*, vol. 30, no. 6, pp. 941–954, 2008.

[7] A. Antonacopoulos and D. Bridson, "Performance analysis framework for layout analysis methods," in *ICDAR*, vol. 2, 2007, pp. 1258–1262.

[8] W. Seo, M. Agrawal, and D. Doermann, "Performance evaluation tools for zone segmentation and classification (pets)," in *ICPR*, 2010, pp. 503–506.

[9] S. Randriamasy and L. Vincent, "Benchmarking page segmentation algorithms," in *CVPR*, 1994, pp. 411–411.

[10] J. Liang, T. I. Phillips, and R. M. Haralick, "Performance evaluation of document structure extraction algorithms," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 144–159, 2001.

[11] F. Shafait, D. Keysers, and T. M. Breuel, "Pixel-accurate representation and evaluation of page segmentation in document images," in *ICPR*, vol. 1, 2006, pp. 872–875.

[12] D. Lopresti and G. Wilfong, "A fast technique for comparing graph representations with applications to performance evaluation," *IJDAR*, vol. 6, no. 4, pp. 219–229, 2003.

[13] A. K. Dasand, S. K. Saha, and B. Chanda, "An empirical measure of the performance of a document image segmentation algorithm," *IJDAR*, vol. 4, no. 3, pp. 183–190, 2002.

[14] S. Mao and T. Kanungo, "Empirical performance evaluation methodology and its application to page segmentation algorithms," *PAMI*, vol. 23, no. 3, pp. 242–256, 2001.

[15] C. Wolf and J. M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *IJDAR*, vol. 8, no. 4, pp. 280–296, 2006.

[16] M. Coteand and A. B. Albu, "Layered ground truth: Conveying structural and statistical information for document image analysis and evaluation," in *ICPR*, 2016, pp. 3258–3263.

[17] T. M. Breuel, "The ocropus open source ocr system." *DRR*, vol. 6815, p. 68150, 2008.